COMP 3000 Operating Systems

Study Session Questions

December 9, 2025

Part 1 - Overview (OSC Ch. 1-2)

1. Which of the following statements are true? Select all that apply. You may also want to justify your choice behing each decision.

    A. Every interrupt necessarily causes a context switch from **one user process to a different user process.**

    B. Direct Memory Access (DMA) allows a device to read/write main memory **without CPU involvement for each transferred word**, but the CPU is still notified when the transfer completes.

    C. The primary reason to have a separate kernel mode is **performance**, not protection.

    D. On a single-core CPU, it is **impossible to have parallelism but still possible to have concurrency**.

    E. Microkernel designs generally achieve lower syscall latency than monolithic kernels because they move most services to user space.

    F. In a layered OS design, a higher layer should not directly invoke services of a non-adjacent lower layer.

2. Place the steps of interrupt handling in the correct order, from the beginning. (1, 2, 3, 4)

    - [ ] the CPU uses the interrupt vector table to find the right interrupt handler
    - [ ] the device controller/software sends the interrupt to CPU
    - [ ] the CPU restores its state and resumes normal operation
    - [ ] the CPU finishes its ongoing instruction and saves current program state

3. Place the following storage systems in order of relative typical speed, from slowest to fastest. (1, 2, 3, 4)

    - [ ] Hard-disk drives
    - [ ] Registers
    - [ ] Cache
    - [ ] Main memory

4. Explain the differences between a hardware interrupt, a trap (system call), and an exception.

    ---

5. Consider the following list of actions. Which of the following should be performed by the kernel, and not by user programs? Select all that apply.

    A. reading the value of the program counter (PC).

    B. changing the value of the program counter (PC).

    C. increasing the size of an address space.

    D. creating a memory segment that is shared between multiple processes.

    E. writing to a memory segment that is shared between multiple processes.

    F. disabling interrupts.

6. Define the terms system bus, device controller, and DMA controller.

7. Explain the difference between multiprogramming and multiprocessing. Include what each term means, what hardware support is required, and an example scenario where each is useful.

Part 2 - Process Management (OSC Ch. 3-5)

8. Which of the following statements are true? **Select all that apply.**
    A. It is possible to have concurrency **but not parallelism**.
    B. Threads within the same process access the same **address space**.
    C. The **Ready** queue contains processes that are waiting for I/O to complete.
    D. In preemptive priority scheduling, starvation of low-priority processes is **impossible**.
    E. A race condition occurs when the correctness of a program depends on the relative timing of threads.
    F. A successful context switch **always** requires saving the state of the currently running process and restoring the state of the next process.

9. What is the use of the return value of `fork()`?

10. Which of the following is shared between threads of the same process? Select all that apply.
    A. integer and floating point registers
    B. program counter
    C. heap memory
    D. stack memory
    E. global variables
    F. open files

11. Under which of the following basic CPU scheduling policies is starvation possible? Select all that apply.

    A. First-Come-First-Served (FCFS)

    B. Shortest-Job-First (SJF)

    C. Shortest-Remaining-Time-First (SRTF)

    D. Round-Robin (RR)

12. What is *context switching*?

13. What is the distinction between a ready thread and a waiting thread?

14. Which of the following statements are true? **Select all that apply.**

    A. With a many-to-one threading model (many user threads mapped to one kernel thread), blocking on a single system call may block all user threads in that process.

    B. Adding more cores **always** decreases execution time for a multithreaded program, as long as there is no I/O.

    C. Round-Robin scheduling is preemptive because it forces a context switch after each time quantum, even if the running process has not blocked or terminated.

    D. In a pure one-to-one threading model, creating a very large number of user threads may stress kernel resources and harm performance.

15. Consider the following set of processes, with the length of the CPU burst time given in milliseconds:

Process	Burst Time	Priority
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

    The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, all at time 0.

    Which of the following algorithms results in the minimum average waiting time (over all processes)?

    A. First-Come-First-Served (FCFS)

    B. Shortest-Job-First

    C. Non-preemptive Priority (with a larger priority number implying a higher priority)

    D. Round-Robin (quantum = 2)

16. A system uses an SRT (Shortest Remaining Time) scheduler. Among processes with the shortest remaining time, the earliest-arriving one is chosen.

    Three processes arrive: P_0, P_1, and P_2. P_0 has a CPU burst of 4 units, P_1 has a burst of $(1+y)$ units, P_2 has a burst of 3 units.

    The processes arrive at times 0, 2, and 5 (one process at each time).

    (a) Determine the arrival order of P_0, P_1, and P_2.

    (b) Find the value of y so that:

    - SRT produces the **same** schedule as FCFS.
    - The average response time of the three processes is **maximized**.

    Response time is the time from arrival to first execution.

Part 3 - Process Syncronization (OSC Ch. 6-8)

17. Match each term to the most accurate and precise defenition.

A. Mutual Exclusion	**1.** Each process waits at most a finite number of turns.
B. Bounded Waiting	**2.** If the critical section is empty, waiting processes must eventually enter.
C. Progress	**3.** Only one process may be in its critical section.

18. Which of the following statements are true? **Select all that apply.**

    A. A binary semaphore can always be used as a mutex, but a mutex cannot always be used as a general counting semaphore.

    B. In a readers–writers lock with reader priority, it is possible for writers to starve even when there is only a single writer repeatedly requesting access.

    C. If a system's resource-allocation graph contains a cycle, the system is definitely in a deadlocked state.

    D. Deadlock avoidance (e.g., Banker's algorithm) can sometimes deny a resource request even though granting it would not immediately guarantee deadlock.

    E. Deadlock detection algorithms **must** know the maximum resource demand of each process to function correctly.

19. (a) How could one PREVENT deadlock in the situation shown below using deadlock prevention techniques discussed in class?

Four buses blocking each other at the Alexander Klellands Plass intersection in Oslo, Norway. Photo dated 24 November 2025.

(b) Provide another (non-operating-system-related) example of deadlock.

20. Given the state below, determine whether the system is in a safe state. If it is safe, provide a safe sequence.

Allocation			
P	R_1	R_2	R_3
P0	2	4	3
P1	0	1	1
P2	5	0	2

Need			
P	R_1	R_2	R_3
P0	3	2	4
P1	1	4	3
P2	2	0	0

Available		
R_1	R_2	R_3
2	1	2

21. Consider the following information about resources in a system:

- There are two classes of allocatable resource labelled R1 and R2.
- There are two instances of each resource.
- There are four processes labelled P1 through P4.
- There are some resource instances already allocated to processes, as follows:
  - one instance of R1 held by P2, another held by P3
  - one instance of R2 held by P1, another held by P4
- Some processes have requested additional resources, as follows:
  - P1 wants one instance of R1
  - P3 wants one instance of R2

(a) Draw the resource allocation graph for this system.

(b) What is the state (runnable, waiting) of each process? For each process that is waiting, indicate what it is waiting for.

(c) Is this system deadlocked? If so, state which processes are involved. If not, give an execution sequence that eventually ends, showing resource acquisition and release at each step.

Part 4 - Memory Management (OSC Ch. 9-10)

22. Which of the following statements are true? **Select all that apply.**

    A. Paging eliminates both internal and external fragmentation.

    B. Multi-level paging trades higher address translation overhead for lower page-table memory usage.

    C. LRU page replacement can be approximated using a single reference bit per page and a circular scan (clock algorithm).

    D. If a process's working set does not fit into memory, increasing the degree of multi-programming will usually reduce thrashing/the # of page faults.

23. Under what conditions might two or more processes share common physical frames? Explain how and why this would be done.

24. If every page in memory is accessed (used) between any two page faults, will the Clock replacement algorithm behave exatcly like FIFO? Justify your response whether true or false.

25. Assume that there are 5 pages, A, B, C, D, and E. Fill in the page reference string and complete the rest of the information in the table below so that LRU is the *worst* page replacement algorithm (i.e., it results in the maximum number of page faults). Use a dash "–" to fill in blank locations. Note that when there is more than one page that is a possible victim, always choose the one with the **lowest frame number**.

Num	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Refs	A	B	C	D	E										
Frame 1	A														
Frame 2	–														
Frame 3	–														
Frame 4	–														
Fault?	X														

26. Consider a virtual memory system that uses 2-level paging. The page size in this system is 256 (2^8) bytes. Each individual page table fits exactly into one memory frame, and the size of each page table entry (PTE) is 8 bytes.

    (a) What is the maximum size (in bytes) of a virtual address space in this system?

    (b) Suppose that there is a process with a virtual address space of the maximum size. How many bytes of memory are occupied by the page tables for this process?

    (c) Suppose that there is a process with a virtual address space of size 10240 ($10 \cdot 2^10$) bytes. Also suppose that the entire address space is in memory. How many valid PTEs will exist in the page tables for this process?

27. Assume that a program has just referenced an address in virtual memory. Describe a scenario in which each of the following can occur. (If no such scenario can occur, explain why.)

    (a) TLB miss with no page fault

    (b) TLB miss with page fault

    (c) TLB hit with no page fault

    (d) TLB hit with page fault

Part 5 - Storage Management (OSC Ch. 11-12)

28. Which of the following statements are true? **Select all that apply.**

    A. SCAN disk scheduling guarantees less total head movement than FCFS for every possible request pattern.

    B. SSD/NVM devices have smaller benefit from seek-optimizing algorithms than HDDs because they have no mechanical heads to move.

    C. For HDDs, logical block addresses (LBAs) are conceptually arranged in order across the disk, even if the controller hides the exact physical layout.

    D. An OS can improve swap performance by spreading swap space across multiple devices instead of putting all swap on a single disk.

29. Match each I/O style to the most accurate definition.

A. Blocking I/O	**1.** A single system call reads from or writes to multiple user buffers in one operation.
B. Non-blocking I/O	**2.** The system call does not return until the requested I/O has completed or an error occurs.
C. Asynchronous I/O	**3.** The kernel starts the I/O and returns immediately; completion is reported later (e.g., via signal, callback, or completion queue) so the caller does not need to poll.
D. Vectored I/O	**4.** The system call returns immediately; if the operation would block, it fails (e.g., with "would block"), and the caller must retry or use readiness notification.

30. Suppose that a disk drive has 8,000 cylinders, numbered from 0 to 7,999. The drive is currently serving a request at cylinder 3,400, and the previous request was at cylinder 3,000. The queue of pending requests, in FIFO order, is:

    120, 3,760, 1,550, 720, 4,090, 2,780, 510, 3,900.

    Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms?

    (a) FCFS

    (b) SCAN

    (c) C-SCAN

31. Which of the following statements are true? **Select all that apply.**

    A. I/O protection is enforced partly by marking I/O instructions as privileged and by protecting memory-mapped device registers from direct user access.

    B. Spooling is used when a device can handle multiple concurrent requests in parallel, so the kernel can overlap operations from different processes.

    C. The kernel's error handling may retry transient I/O errors and can decide to stop using a device that exhibits too many correctable errors.

    D. The kernel's device-status table tracks the state of each I/O device so drivers and the kernel can coordinate ongoing operations.

32. Consider the following I/O scenarios on a multitasking workstation:

    (a) A USB keyboard used with a text editor and graphical terminal.

    (b) A network printer shared by many users, connected over Ethernet.

    (c) An SSD that stores the operating system and user home directories.

    (d) An audio output device (sound card) playing compressed music or video sound, using DMA.

    For each scenario, would you design the operating system to use buffering, spooling, caching, or some combination?

    Briefly justify your choices.

Part 6 - File System (OSC Ch. 13-14)

33. Which of the following statements are true? **Select all that apply.**

    A. Sequential file access requires the hardware to support sequential-only disks; the OS cannot simulate sequential access on random-access devices.

    B. Direct (random) access to a file is commonly implemented using relative block numbers so the OS can choose where to place blocks on disk.

    C. In a two-level directory scheme, all users share a single directory without separation, which avoids name collisions between users and supports hierarchical grouping of their files.

    D. The OS understands the meaning of file extensions like `.c` or `.docx` and uses them to decide how to interpret file contents.

    E. In an acyclic-graph directory, a file or subdirectory may have multiple names (aliases), which can lead to dangling references if one alias is deleted.

    F. General graph directory structures must use additional mechanisms, such as forbidding links to directories or running cycle detection, to avoid cycles.

34. Explain how directory structures (flat, two-level, tree, acyclic, graph) affect usability, protection, and sharing. Give an example scenario where each structure would be particularly convenient or problematic.

35. Explain the main differences between using FSCK and using journaling for crash recovery. Under what circumstances might you still run FSCK on a journaling file system?

36. Which of the following statements are true? **Select all that apply.**

    A. In FAT-based allocation, the file's directory entry stores pointers to all its data blocks directly, so the FAT table is only used for free-space management.

    B. Unix inodes combine several direct block pointers with one or more levels of indirect pointers to efficiently support both small and large files.

    C. A unified buffer cache avoids double-caching by using a single cache for both page-based memory-mapped I/O and traditional file system I/O.

    D. Synchronous writes always go through the buffer cache and can be delayed there for performance, as long as metadata remains consistent.

    E. In write-ahead logging, metadata changes are written to their home locations first, and only afterwards are they recorded in the journal.

    F. Because a journal log is finite, most journaling file systems treat it as a circular buffer and eventually reuse old log space.

37. Match each file allocation method with the description that best fits its behavior and trade-offs.

A. Contiguous allocation	**1.** The file system records only a starting position and a length for each file. Accessing the k-th block is very fast once the start is known, but accommodating file growth may require relocating the file or adding special "extension" regions, and free space can become fragmented over time.
B. Linked allocation	**2.** Each block used by a file contains extra metadata that tells the system where to find another block of the same file. Files can easily grow one block at a time without moving existing data, and free space can be used in small pieces, but jumping directly to the k-th block may require visiting many other blocks first.
C. Indexed allocation	**3.** Information about where a file's blocks live is collected in a separate structure that stores many block references together. This allows the system to compute the location of the k-th block with one or a few lookups, even if blocks are scattered, but it consumes additional space for that structure and may be wasteful for very small files.

38. Consider a file system that uses inodes to represent files. Disk blocks are 2 KB in size, and a pointer to a disk block requires 4 bytes. Each inode contains 6 direct block pointers, as well as one single-indirect and one double-indirect block pointer. A *single-indirect* pointer refers to a block that contains only block pointers, and a *double-indirect* pointer refers to a block that contains pointers to such single-indirect blocks.

    What is the maximum size of a file that can be stored in this file system?